The Report Committee for Alan Aaron Bernstein
Certifies that this is the approved version of the following report:

Foveated Coding for Persistics

APPROVED BY
SUPERVISING COMMITTEE

Supervisor: _____

Alan C. Bovik

_____

Robert W. Heath

Foveated Coding for Persistics

by

Alan Aaron Bernstein, B.S.E.E.

Report

Presented to the Faculty of the Graduate School
of the University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science in Engineering

The University of Texas at Austin
December 2012

Acknowledgements

I would like to thank all those involved in this project for their assistance and sponsorship, including my advisor Alan Bovik, reader Robert Heath, my colleagues at LIVE and WNCG, Sheila Vaidya, Holger Jones, Mike Goldman, and others at Lawrence Livermore National Laboratory.

Foveated Coding for Persistics

by

Alan Aaron Bernstein, MSE
The University of Texas at Austin, 2012
SUPERVISOR: Alan C. Bovik

Persistics is an advanced framework for processing wide-area aerial surveillance video. This framework handles the tasks of data collection, stitching of multi-sensor imagery, image registration and stabilization, motion tracking, and compression. As the technology for image sensor sizes improves, significant improvements in compression techniques are necessary in order to make full use of the data. Because the information of interest in such video is naturally moving, point-like targets, the applicability of foveated coding to the compression problem is an interesting question. Foveated coding, a compression technique that was designed to be perceptually optimal for the human visual system, has several components that are appropriate to the persistics compression problem. Foveation is applied in several different scenarios and methods to persistics data. As foveation can make good use of the persistics tracker data, a problem affecting tracker performance is explored as well. The multi-sensor stitching component of persistics can generate artifacts that reduce the effectiveness of the tracker. A method for characterizing, detecting, and correcting such artifacts is desirable. These three concepts are explored, and a method for detection is developed. Components of these algorithms were absorbed into a more general framework for artifact correction.

Contents

1 Introduction

1.1 Persistics

Aerial surveillance video of human activity is an important asset for military and intelligence organizations. Increasing image sensor resolutions result in the familiar explosion in data rates, which is a problem for both compression and storage. For applications where a near-real-time video link from an aerial platform to a viewer is desired, the compression problem is increasingly difficult, in part due to limited computational power. A distinct problem is that of a video viewing client requesting video sequences from a remote server. This second scenario may have more computational power available, but in practical application more functionality is required, as a viewer should have free, bidirectional control over space, time, and scale of the video display.

Within the last several years, imagery and drone technology has improved to the point that a single aerial platform is capable of monitoring tens of square kilometers, with a resolution sufficient to identify vehicles and individuals, for hours at a time (See Fig. 1). This promises a new level of functionality, but the ability to achieve high compression rates on this data is crucial. Persistics, developed by a group at Lawrence Livermore National Laboratory, is a framework upon which such compression can be built.

Persistics consists of a pipeline with several different components for preprocessing imagery. Image stitching, registration and stabilization are the most relevant steps here. Because the platform uses an array of optical elements, each with an identical array of small image sensors, a stitching step is necessary to combine the small number of large frames into a single, larger frame. Because the platform fixates on a single point by following a circular flight path while rotating the camera viewing axis, a stabilization step is necessary to produce a video sequence with the appearance of a static recording position. Ultimately, the desired output is a single video sequence, combining the imagery from all sensors, that has been spatially stabilized, so that any individual pixel represents the same approximate ground location, throughout the sequence.

Even after the basic persistics processing is completed, the imagery retains a number of artifacts. Some, like natural lighting variations, parallax effects on tall structures, or specular reflections, are unavoidable. Others, like stitching errors, are essentially due to algorithm flaws. All of these types of artifacts must be handled in their own way.

The source of the imagery used in this work is several domestic test flights of the

Constant Hawk and ARGUS aerial platforms, using monochrome image sensors varying in size from 66 megapixels to 1.8 gigapixels, covering up to 16 square kilometers at a frame rate of 1Hz or 2Hz. The platform has the capability to fixate on a static ground target for hours at a time, although the video sequences used here were shorter in both duration and in spatial extent.

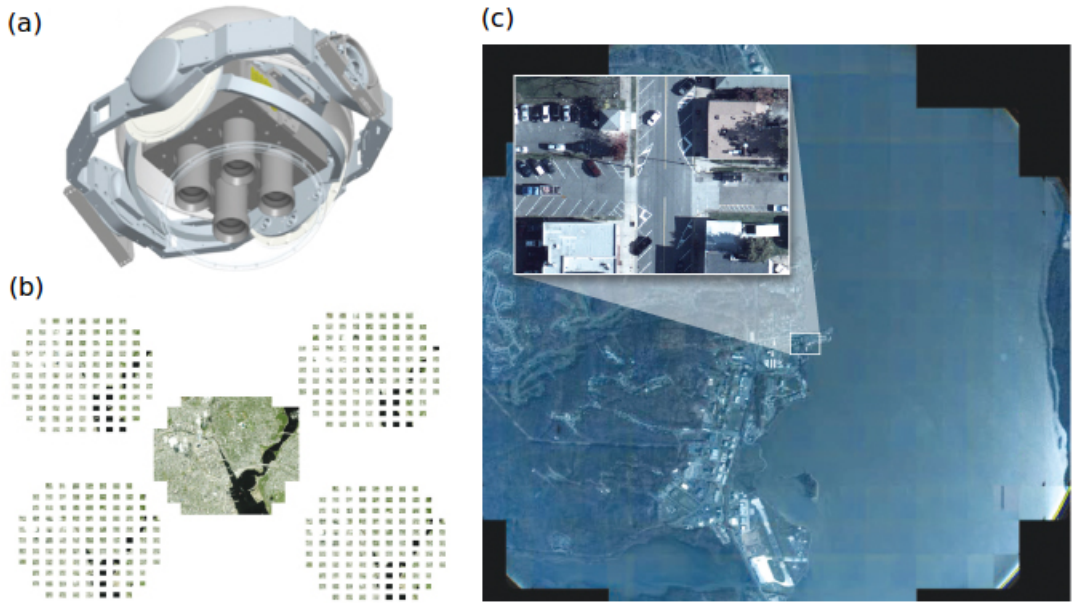

Figure 1: (a) Example of image sensor hardware configuration. (b) Output of individual small sensors before stitching. (c) Full frame with inset showing individuals and vehicles visible within an aggregate wide-angle view. Adapted from "From Video to Knowledge," Science and Technology Review, April 2011. Web. 5 Dec. 2012.

1.2 Foveated Coding

Foveated coding is a compression technique directly inspired by the human visual system (HVS), and built on top of the set partitioning in hierarchical trees (SPIHT) algorithm (4). The term "foveated" refers to the density profile of light sensing cells (cones) on the retina; human vision has a low-resolution characteristic in the periphery, with a much higher resolution in the center, or fovea. Note the qualitative similarity between the two cone density curve and foveation profile curve in Fig. 2. In human vision, the foveated characteristic is supplemented by quick motions of the eye, combined with processing that effectively retains the high-resolution information even after the eye's target changes.

In foveated coding for video, it is possible to use hardware that mimics the eye motion of the HVS, thus reducing the data rate directly by reducing the number of light sensing pixels. This approach has the drawback of requiring significant knowledge of the position of the object or objects of interest in the scene, so for this application a software-based approach is preferable. In this approach, the full resolution imagery is passed to a system that first determines a foveation point (or region), then applies a foveation profile to the imagery, which is used to encode it in a way that prioritizes information near the foveation point.

The foveated coding process itself is well understood; the challenge is in determining a foveation region for a given scene. In [2], a relatively simple method for finding foveation regions, for specific types of scenes, is described. In this work, much of the focus was on developing a method for finding foveation regions. To this end, several different extensions to the basic point-like foveation were considered.

A core component of foveated coding is its basis in the properties of the HVS. The foveation profile is carefully designed to match that of the typical HVS characteristics, based on experiments with human subjects, and simple statistical modeling for viewing geometry. This basis in physiology distinguishes foveated coding from region-of-interest coding, in that foveated coding is perceptually optimal in a certain sense. While this is an important aspect of foveated coding, its importance in this work varied depending on the scenario. The capability for a type of region-specific successive refinement transmission was more broadly relevant.

This work was sponsored by Lawrence Livermore National Laboratory, with the goal of finding a place for foveated coding within the persistics processing pipeline. The work focused on several different, related goals. Originally motivated by the desire to use foveated coding to reduce the data rate required over the platform's RF link, the work then shifted to the problem of compression for archival storage, as well as the problem of compression for transmission from a ground-based server to a viewing client. Some time was also spent on the problem of removing artifacts resulting from image stitching, a preprocessing step that is important for ensuring that the persistics target tracking system works correctly.

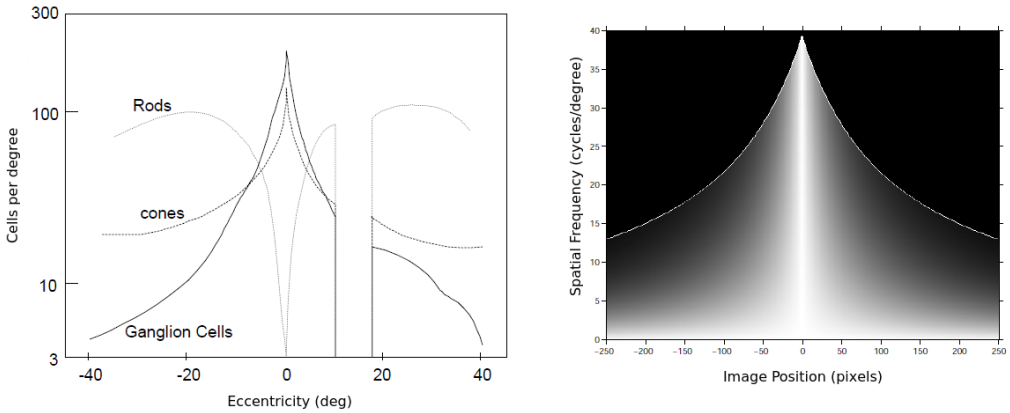

Figure 2: Left: Model for density of light sensing cells and ganglions in the human eye. Right: "Importance" profile for foveated coding, based on contrast sensitivity of the HVS, which is closely related to the density of cells in the eye. Adapted from "Embedded Foveation Image Coding," IEEE Transactions on Image Processing, Vol. 10, No. 10, October 2001.

2 Background

2.1 Stitch Correction

Due to constraints in the persistics system, a number of practical issues in the data must be corrected before it is usable by downstream components. One significant problem is an artifact in the stabilized imagery referred to as seams (equivalently, stitches, sensor boundaries, etc., see Fig. 3). These are the result of an insufficient stitching algorithm; when the imagery from multiple sensors is merged incorrectly, this is one possible artifact. The *stitch correction problem* encompasses the correction of all artifacts that are the result of the stitching process, regardless of the exact character or cause. Several distinct types of seam errors were observed in the data, with varying complexity, both in terms of characterization and the required correction techniques.

Figure 3: Simple seam artifact example. Left: near-vertical seam visible in left-center of image, after manual stitching. Right: seam eliminated after ideal dense image correspondence. Adapted from "From Video to Knowledge," Science and Technology Review, April 2011. Web. 5 Dec. 2012.

Persistics uses pixel-level dense image correspondence to stitch frames together. However, it was not always possible to obtain the original data captured on the platform before it had been processed by an alternative pipeline at a different organization. In these cases, the visible seam errors consisted of overlap between the sub-frames of two or sensors, of varying length, width or angular dispersion, and transparency. The main distinction in the types of errors was the complexity of the boundary between different frames. In the most benign cases, this boundary was a single, sharply defined, straight line boundary between two sensors. One additional level of complexity is the case of a degree of transparency in a

thin, wedge-shaped overlap region. Another level of complexity is the case of three or four frames overlapping in a simple way, with overlap regions radiating from a single, mutual vertex. Finally, in the most complicated case, multiple frames may mutually occlude each other in such a way that the overlap regions are not simple lines, or wedge-shaped regions, but complex combinations of short lines (see Fig. 4). This occurs when the stitching algorithm determines depth maps for each sub-frame, based on physical reference data, and these depth maps are mutually occlusive.

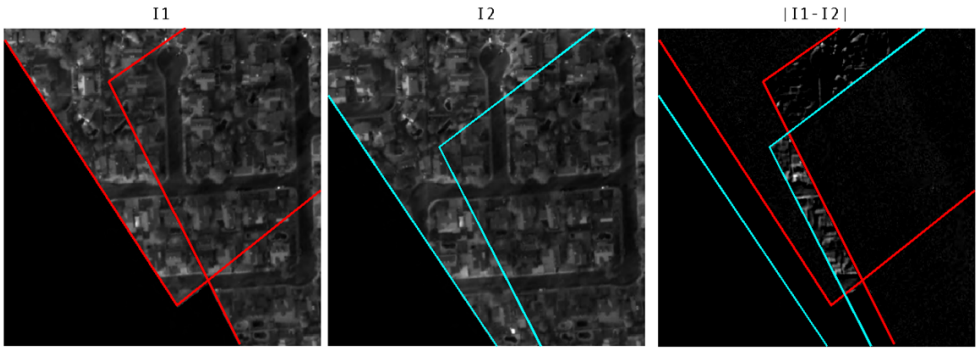

Figure 4: Complex seam artifact examples. Left: Frame N with corner seams. Center: Frame $N+1$ with corner seams. Right: corner seams overlaid.

Because the persistics object tracker is sensitive to these seam artifacts, correction or reduction of the artifacts is an important preprocessing step. The character of these seams can vary unpredictably between these cases, so a framework was developed to intelligently handle different types of seam artifacts.

2.2 Foveated Coding for Persistics

Within the context of persistics, we identified three potential applications of foveated coding, and two basic methods for applying foveation. Although focus shifted between these applications over the course of the project, the functionality of the compression algorithm remained similar.

Several minor goals were identified as well: The spatially-variant nature of SPIHT coding is used in foveated coding to model the sensitivity of the HVS, but it can be used more generally as well. In particular, our imagery contained several spatially varying distortions or effects, which can be interpreted as a sort of redundancy. This redundancy can be reduced with SPIHT-like coding.

A vignetting profile of the lenses was found to have an intensity rolloff of up to a factor of two near the sub-frame edges. This means the dynamic range for the imagery was notably lower over a substantial area, by up to about one bit. After a correction factor is applied to the image, the lowest order bits in these regions contain no useful information, so our SPIHT-like coding should assign zero importance to these. Similarly, ground sample distance (GSD) distortion, the physical ground size corresponding to image pixels, varies widely depending on the viewing angle of the camera. This can be also be represented as a redundancy of data, and discarded in a similar way.

2.2.1 Different scenarios

The original motivation for investigating novel compression techniques in persistics came from the challenging problem of achieving high compression rates (1000:1 and higher) in the RF link from the aerial platform to the ground station. This scenario is the most challenging of the three, since it involves permanently discarding data. Loss of any data that includes motion of individuals or vehicles is undesired; minimizing false negatives is crucial. The computational requirements for ensuring such performance while achieving high compression rates were found to be infeasible, and so this scenario was abandoned in favor of a simpler one.

Another possibility, known as the "archival" scenario, is less challenging in that it may be operator-aided, or repeated with various compression rates until a desirable result is found. In this scenario, the full video sequence is retrieved after the platform lands, then passed through the foveated coding system, to reduce size for long-term storage. Finally, the "server-client" scenario is the least complex of the three. In this case, the full video sequence, or a high-quality compressed sequence, is stored on a server. Analysts operate remote viewing stations, requesting data over relatively small windows in time and in space. Since no data need be permanently discarded in this case, false negatives are a less significant error. Most of the work focused on these latter two scenarios after the former was found to be infeasible.

It is notable that the third case has the most to benefit from the perceptual aspect of foveated coding. Since the viewing geometry of the user is known fairly accurately, very specific parameters for the foveation profile can be used to achieve their original purpose of approximating a perceptual optimum.

2.3 Target Foveation

The core idea of foveated coding is to select a foveation point in a scene, and encode the image in such a way that priority is given to the region around the foveation point in a perceptually efficient manner. This idea lends itself to extension for this application in a few different ways.

Although the perceptual optimality of foveated coding is an important component of the original work, its applicability here is limited, depending on the scenario. For example, when compressing for a server-client transmission, the viewing geometry might be known fairly accurately. In this case it would be desirable to set the foveation profile parameters based on that geometry, and achieve a viewer-specific perceptually optimal display. However, when compressing for transmission over the RF link, or for archival storage, the viewer geometry may not be known at all. In this case, although the perceptual optimality of foveated coding is still useful, the perceptual quality of the compression is not as sensitive to foveation profile parameters.

The implication is that the foveation profile parameters can supply an additional degree of freedom, by modifying the "importance map", so that the algorithm here is a sort of hybrid between foveated coding and SPIHT. Profile parameters can be varied to effectively modify the compression rate directly. This is an important factor for compensating for a widely varying foveation region area. Foveation region area (angular width) for the human visual system is not variable, but one simple extension of the basic idea of foveated coding allows for this. The system can select multiple foveation points, and apply foveation separately to all of them. For a given foveation profile setting, the result is an increase in foveation region area of a factor equal to the number of foveation points.

While this approach is a clear necessity for a system capable of surveillance of multiple individuals or vehicles, it is not the only possibility, so this concept is a necessary first step. The "Foveated encoder (known targets)" block in Fig. 5 illustrates the location of this type of foveation step in a coarse block diagram of the overall persistics pipeline.

Two different methods for determining target locations were considered. The use of persistics tracker data is the most direct; since the persistics system already includes a tracker, simply passing its target tracks to the target foveation system should work well. However, the persistics tracker is computationally intensive, so a simpler tracker was also developed for the same purpose.

2.4 Road Foveation

One alternative to the multiple target-foveation approach is based on the notion of "road foveation". Because the goal of compression here is to minimize false negatives, it is of interest to preempt the discarding of targets. However point targets are chosen, some fraction of them will be missed, so any method of preventing this data from being discarded is desirable. The concept of dividing the image into road and non-road regions was pursued for this purpose. Vehicles are not expected to travel on roads exclusively. However, they do return to roads often enough that retaining road region video should help to retain more targets.

This approach suffers a problem similar to the target foveation approach: the locations of roads must be known beforehand. Again, with target tracks, road networks can be estimated, but since the tracker is expensive to run, it is not necessarily available. This suggests the alternative of a cheaper road-detection system.

Given that both point-target foveation and road-network foveation can be used, the next step is to combine these two approaches. Since each approach results in an importance map that is then passed the encoder, a combined importance map can be found as a simple convex combination of the two.

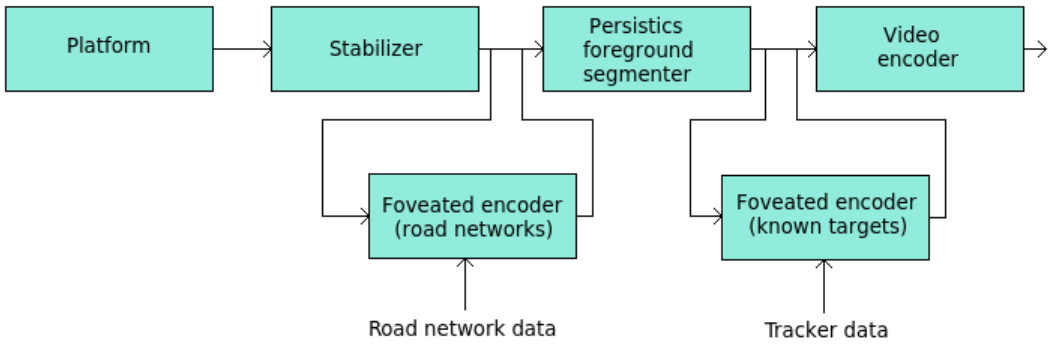

Figure 5: Stitch detection block diagram. Lower blocks illustrate insertion of two different types of foveated coding into the persistics pipeline.

3 Research

3.1 Stitch Detection

The method for stitch correction described here consists of a sequence of steps, divided into two main parts: detection and correction. The detection process progressed through a number of iterations, finally settling on a user-guided approach that can be applied in an intelligent way across an entire video sequence with minimal interaction. As this process was integrated a larger, development pipeline, this work focused on two of the blocks in this diagram, the "refine seam segments" and "generate refined seam map" blocks.

The first of these consists of a series of progressively finer estimates of seam locations, described as follows, and illustrated in Fig. 6:

1. Estimate 6 "exterior seam endpoints" and 2 "interior endpoints" using bounding polygon (Fig. 6a.).
2. For each exterior endpoint, search for peak in local difference subimage, use this as second estimate of exterior endpoint (Fig. 6b.).
3. Process region near second estimate with Hough transform for third estimate (Fig. 6c.).
4. Process K regions along third estimate with Hough transform to produce $2(K + 1)$ estimates for each seam; find consensus or highest confidence among these to produce fourth estimate, along with confidence value for each seam. (Fig. 6d.).

Given a set of detected stitches, correction is the next step. Initially, a SIFT-based approach was used for this, but the correction was later handled by a downstream algorithm. This SIFT approach was explored briefly; it consisted of finding a least squares estimate of a homography based on the two disparate sets of SIFT keys. This homography was then applied selectively to each sub-frame region.

3.2 Foveated Coding

The foveation process itself is relatively mature, so research focused on methods for determining foveation points. The persistics system includes a motion tracker, so this was considered as a possibility for providing target-based foveation points. As this tracker is computationally intensive, it runs on a computing cluster that requires site access, so development on this approach was slow and difficult.

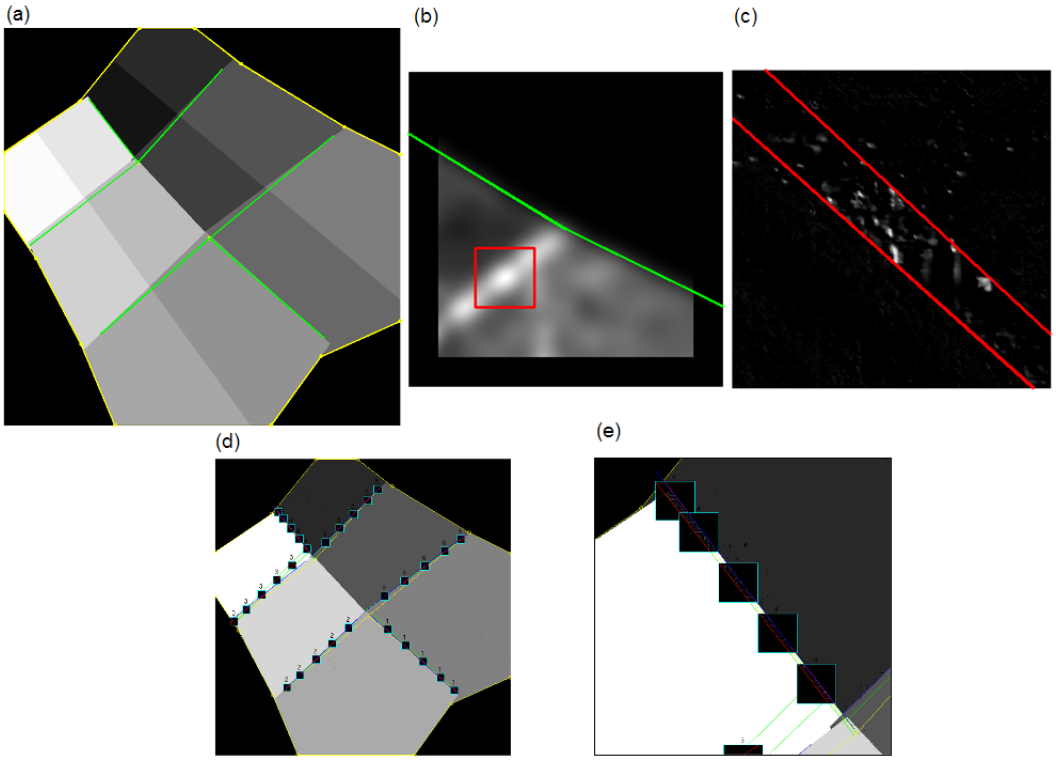

Figure 6: Steps in stitch detection process. (a)-(d): steps 1-4. (e): zoom of final result.

A more usable method of foreground/background separation was developed, based primarily on the RANSAC algorithm. Due to the relatively slowly-varying nature of the background in our video, a cubic polynomial fits most background pixel intensities over time fairly well. RANSAC was used to estimate such a polynomial, for each pixel, in time windows that cover approximately 90 deg of one cycle around the platform's fixation point. A modification of RANSAC was developed to determine outliers by iteratively choosing the best $P\%$ of points that fit a current polynomial estimate. These outliers represent foreground pixels, while the pixel intensities that fit the model curve represent background pixels. This single step produced a reasonably high-quality foreground/background separation for our test data. Including steps to reject small foreground regions, extremely fast regions, or periodic or immobile regions improves the separation enough that it can be used as a tracker. Note that this method was effectively a proxy for the real persistics tracker system; since the real tracker was not available for arbitrary test video sequences, tracker data had to be generated using a reasonably fast method.

These tests were performed on three disparate subsets of video sequences from two

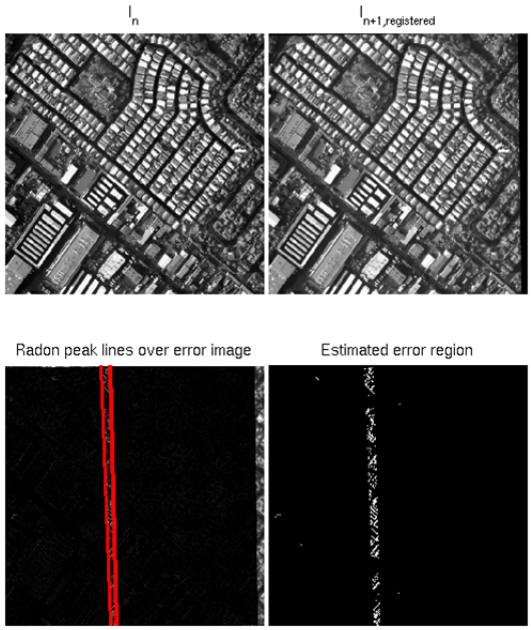

Figure 7: Detail of second step in stitch detection process, showing difference image and initial estimated seams.

different flight tests. One subset was chosen for its empty fields, allowing an example of high compression levels. A second subset was chosen for its dense roads and residential neighborhoods, allowing an example of high activity. The third subset was chosen as a balance between these two, with both types of regions present in the same spatial region.

The process of applying the foveation based on road networks, or point-like targets, is illustrated in Figures 8-12. This process is straightforward, and described conceptually in the previous section. In either case, essentially an importance map is generated, and passed to the FSVC video encoder. For multiple point-like targets, the map is generated by simply summing the basic foveation profile centered at multiple points. For the road networks, with line-like foveation regions, an extruded foveation profile is applied along the length of the line.

In the case of the road network mask, this map is generated by extracting road network estimates from the target tracks. This is only an estimate of the road network, but is still acceptable for applying foveation, due to the rolloff of the foveation profile. In the case of the point-like target importance map, the raw tracker data is used, and foveation is applied to the target track positions. The third possibility, a linear combination of these two methods,

represents a sort of tradeoff between certainty of target locations and higher compression rates: if the system's estimates of target locations can be trusted, then more weight can be given to the point-like foveation regions, which increases the relative bandwidth spent on image data near targets. If these estimates are not trustworthy, then more weight can be given to the line-like foveation regions, presenting a better chance of allowing targets to move into the foveation regions.

Target degradation is shown in Fig. 12. These sequences illustrate that the performance of this foveated coding approach is acceptable up to a compression rate of around 100. After this point, the target structure seems to degrade to an unacceptable quality level. Using this as a proxy for the desired performance metric of the tracker performance, we can conclude that the tracker will work well below the compression rate of 100, but likely poorly above it.

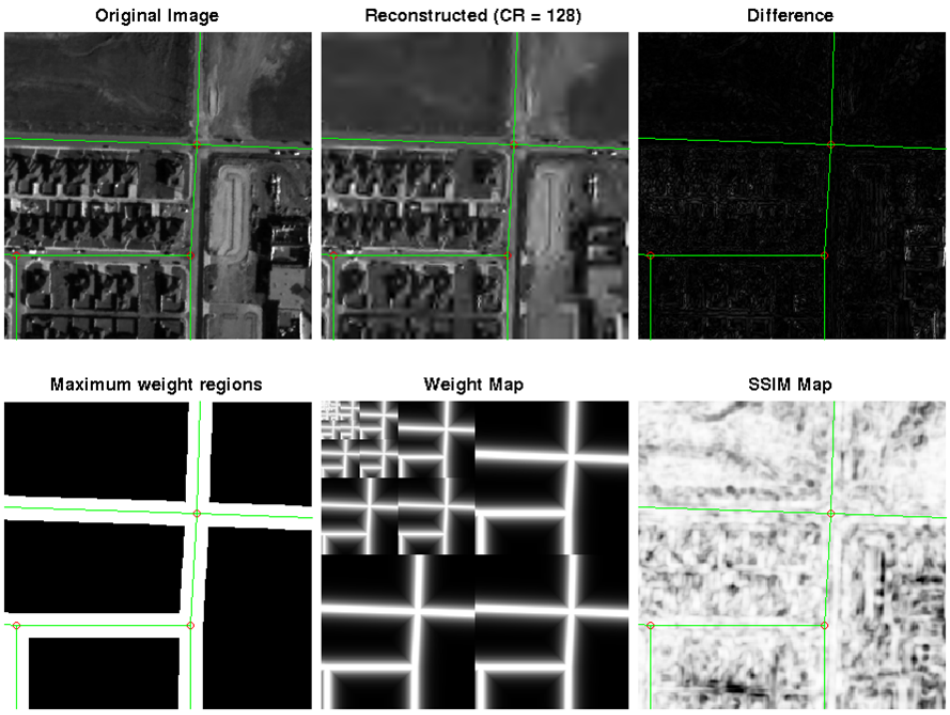

Figure 8: Illustration of road foveation.

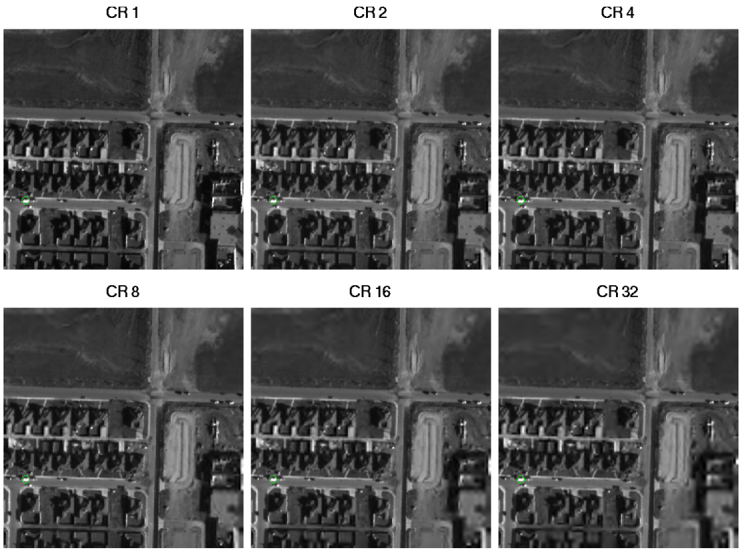

Figure 9: Comparison of image quality vs. compression rate for road foveation. Degradation is not very apparent at this scale.

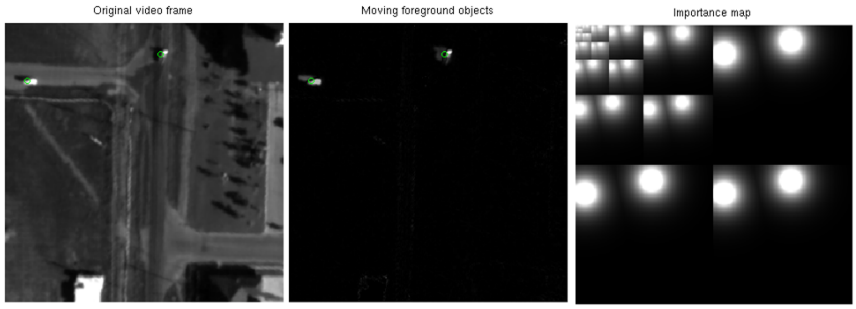

Figure 10: Illustration of target foveation.

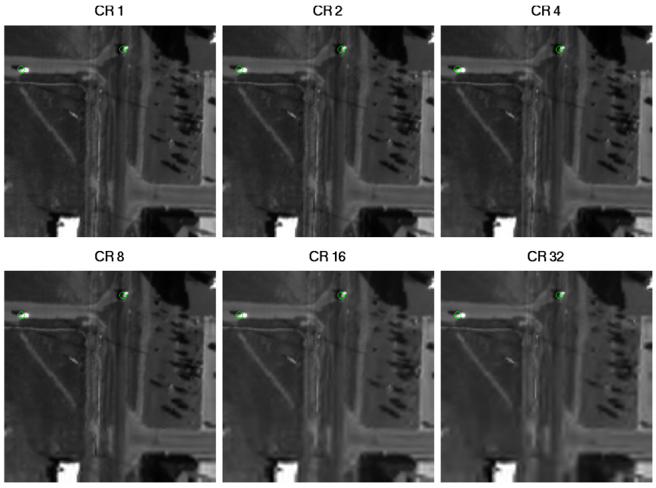

Figure 11: Comparison of image quality vs. compression rate for target foveation, for a subset of a single frame from the encoded sequence. Again, degradation is not very apparent here.

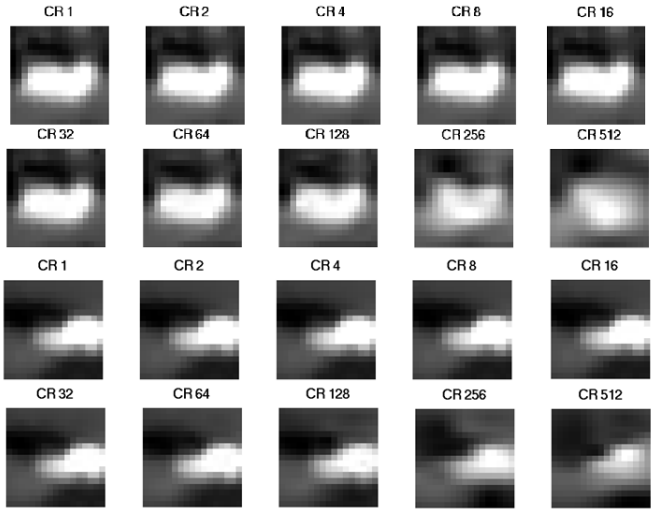

Figure 12: Comparison of zoomed target image quality vs. compression ratio for both road and target foveation. Top sequence of 10: road foveation. Bottom sequence of 10: target foveation. Again, degradation becomes apparent around a compression ratio of 100, although the target appears to retain its structure at a slightly higher compression ratio, compared to road foveation.

4 Results

4.1 Stitch Detection Results

Examples of stitch detection final results are shown in Fig. 13. The six quadrilateral regions represent the ground truth data; each distinct gray level indicates that a pixel of that intensity has been mapped to one particular sub-frame. This ground truth data, available for only a small subset of imagery, allowed for verification of the stitch detection methods. The detected seam edges are highlighted with red lines. The estimates on the left, representative of the more common case, shows all six estimated seams closely matching the true seam locations, accurate to within a few pixels for most of the lengths of the seams. The estimates on the right illustrate a particulary poor failure; the estimates are clearly not close to the true locations. In this example, the failure is likely due to the presense of sharp edges in the road network.

A summary of the overall stitch detection performance is shown in table 1. For each of eight full frames, an estimate was made for each of the six external seams. Most of these estimates were accurate, although a high percentage of them were accurate for the *next* frame in the sequence, rather than the current frame. The cause of this error is unclear. More problematic are the 5/55 estimates that were simply incorrect; this is an area for future investigation.

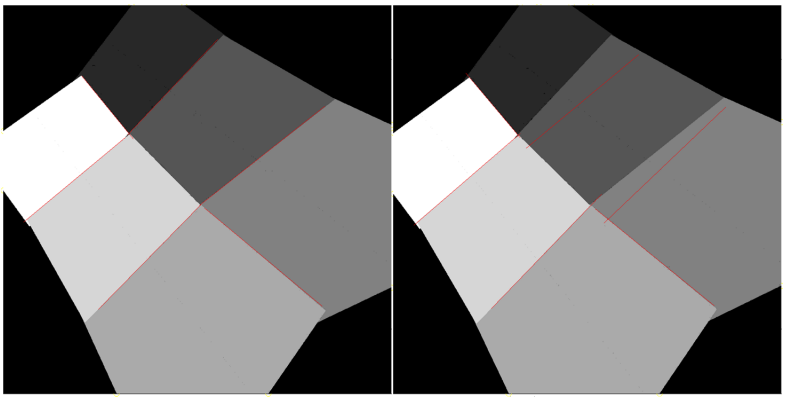

Figure 13: Stitch detection results. Left: accurate estimates for all seams. Right: inaccurate estimates for two seams.

de

seam	1	2	3	4	5	6	total	total%
# correct, current image	4	7	4	9	3	1	28	52%
# correct, adjacent image	5	1	4	0	5	6	21	39%
# totally incorrect	0	1	1	0	1	2	5	9%

Table 1: Performance of seam detection, showing 91% success rate for estimation of the six seam locations, separating six sub-frames, on each of 8 full frames.

4.2 Stitch Correction Results

Although work on stitch correction was quickly handed off to the downstream algorithm, some progress was made. Fig. 14 shows the SIFT keys detected on two frames, used to compute the homography that is then applied to one of the two regions. This method needed work still, but did give reasonable results for the small data set upon which it was tested.

The stitch detection and correction process developed in this work was shown to be a suitable method upon which to build a framework for more general and automated correction. While the correction method was replaced with a more robust algorithm, the detection step was incorporated into a larger framework, which included several different methods for estimating seam locations, a manual adjustment and learning component, and a database of seam artifact types, used to help determine the right type of correction method.

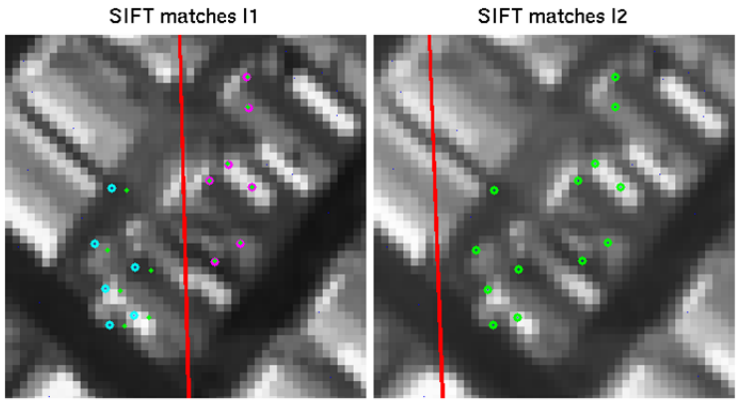

Figure 14: Stitch correction results. Left: SIFT keys for frame N, showing mismatch relative to frame $N+1$, with SIFT keys for frame $N+1$ overlaid. Cyan/Magenta denote the side of the seam that the key falls on. Right: SIFT keys for frame $N+1$. The seam has moved a noticeable distance between the two frame times.

4.3 Foveated Coding Results

The results of the foveated coding approach to persistics video compression are summarized in the performance curves of Figures 15 and 16. SSIM and MSE are used as two approximations to the desired performance metric, the tracker performance. With the tracker output unavailable, these proxies for the tracker performance showed consistent results across the three subsets of data.

The curves for both road-based foveation and target-based foveation look similar, but there is a somewhat significant difference between these two cases. At the point on the x-axis where the SSIM curve begins to decline, the drop is sharper and slightly earlier for the road-based foveation, compared to target-based. This is moreso true for the quality of the target region, which matches the performance seen in the subjective image quality of the target in Fig. 12. This indicates stable performance, with a consistent compression rate of near 100.

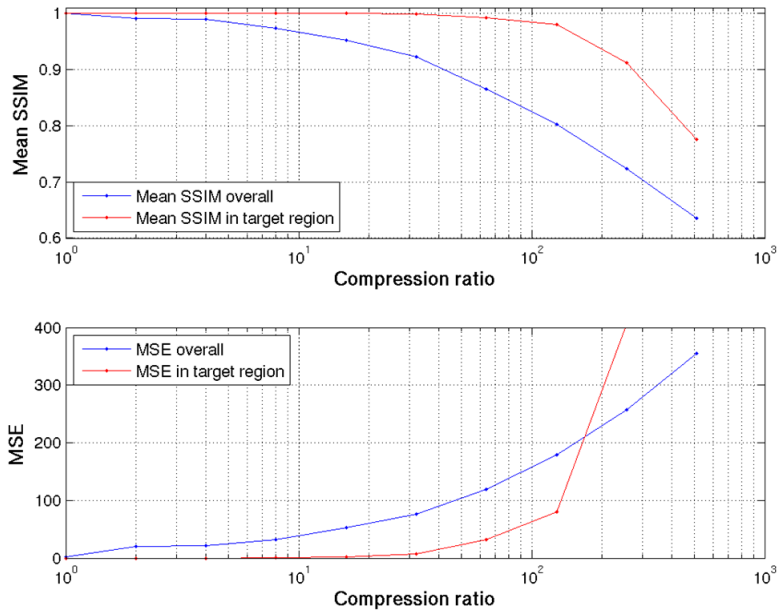

Figure 15: Degradation vs. compression ratio for road-based foveation. SSIM is used, as it is known to correlate well with perceptual image quality.

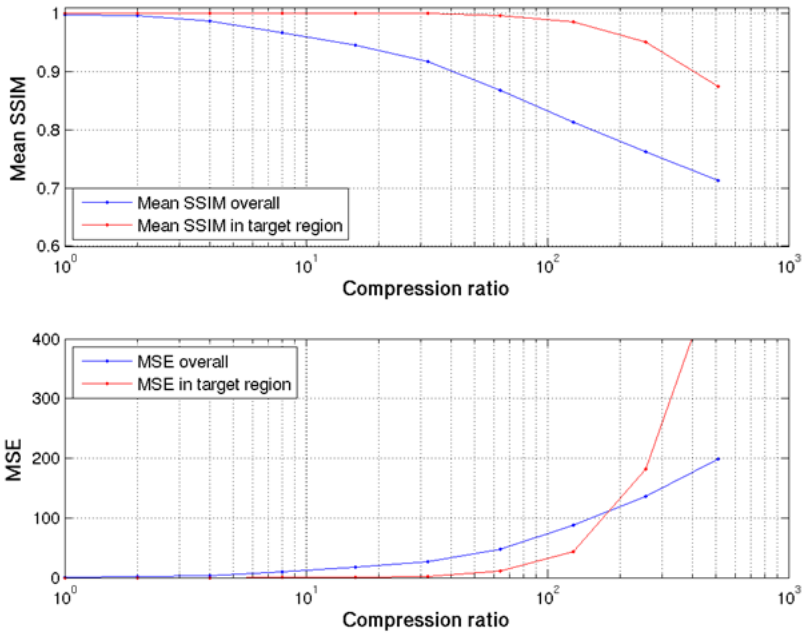

Figure 16: Degradation vs. compression ratio for target-based foveation.

5 Conclusion

5.1 Stitch Correction

Although the full stitch correction problem (including correction of all types of stitching artifacts observed thus far) was ultimately found to be out of the scope of this work, some progress was made on the specific case addressed. The investigation into the full problem suggested a number of interesting directions for future work.

The most obvious next step is the improvement of the various stitching processes used on the original imagery, although this is not always available to the persistics pipeline. Given that these artifacts will necessarily be found in the data, another important step is a more complete characterization, detection, and correction of these artifacts. In lieu of eliminating the problem from the data altogether, the best solution would be an automated system for both detecting and correcting these artifacts.

The problem of characterizing stitching artifacts was an interesting direction. This topic was discussed briefly, but soon abandoned as too tangential to the original task at hand. This question has ample room for further investigation. However, the immediate utility to the persistics system is not clear.

5.2 Foveated Coding

Although foveated coding was ultimately not adopted into the persistics system, the results of this study showed that the method is well-suited to the stabilized aerial video compression problem. The persistics system suffers from a need for extremely high compression rates, and the approach used to apply foveated coding here was not able to achieve such rates. In future work, if a similar problem with less extreme constraints is presented, a similar approach should be applicable.

There are several clear avenues for further investigation for the foveated coding aspect of the work. Unfortunately, because foveated coding is not used in persistics, these topics may not be investigated fully. The most obvious topic is simply the improvement of the foveated coding compression rates, possibly to be achieved with a combination of better estimates of target and road locations, and application-specific modifications to the SPIHT-like importance map profiles.

Another direction is the development of the simple target tracker developed here.

This investigation produced a useful algorithm. Not only is the simple target tracker a reasonable algorithm for providing rough estimates of point-like target locations, but the outlier-rejecting modified RANSAC algorithm is a useful algorithm in itself, suitable for a wider range of model-fitting problems. Either of these components of the simple tracker could be improved upon. The simple tracker still presents a useful model for persistics, although it is certainly applicable to other stabilized aerial tracking problems as well.

6 References

[1] Z. Wang and A.C. Bovik, "Embedded Foveation Image Coding," IEEE Transactions on Image Processing, Vol 10, No 10, October 2001.

[2] Z. Wang, L. Lu, and A. C. Bovik, "Foveation Scalable Video Coding with Automatic Fixation Selection," IEEE Transactions on Image Processing, Vol. 12, No. 2, February 2003.

[3] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," IEEE Transactions on Image Processing, Vol. 13, No. 4, April 2004.

[4] A. Said, and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," IEEE Transactions on Circuits and Systems for Video Technology, vol. 6, no. 3, pp. 243-250, June 1996.

[5] Z. Wang, and A. C. Bovik, "A Universal Image Quality Index," IEEE Signal Processing Letters, Vol 9, issue 3, pp. 81-84, March 2002.

[6] D. G. Lowe, "Object Recognition from Local Scale-Invariant Features," Proc. of the International Conference on Computer Vision, September 1999.

[7] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," Comm. of the ACM 24 (6): 381395, June 1981.

[8] Y. Lai and C. J. Kuo, "A Haar Wavelet Approach to Compressed Image Quality Measurement," Journal of Visual Communication and Image Representation 1, 17-40, 2000.

[9] A.B. Watson, G. Y. Yang, J. A. Solomon, and J. Villasenor, "Visiblity of Wavelet Quantization Noise," IEEE Transactions on Image Processing, Vol. 6, No. 8, August 1997.

[10] G. Vesom, "Pixel-wise Motion Detection in Persistent Aerial Video Surveillance," Computer Vision and Pattern Recognition Workshops, 16-21 June 2012.

[11] A. Heller, "From Video to Knowledge," Science and Technology Review, April 2011. Lawrence Livermore National Laboratory, 5 December 2012 <https://str.llnl.gov/AprMay11/vaidya.html>.

[12] J. Hu, et al, "Road Network Extraction and Intersection Detection From Aerial Images by Tracking Road Footprints," IEEE Transactions on Geoscience and Remote Sensing, Vol. 45, No. 12, December 2007.

[13] M. Amo, F. Martinez, and M. Torre, "Road Extraction From Aerial Images Using a Region Competition Algorithm," IEEE Transactions on Image Processing, Vol. 15, No. 5, May 2006.

[14] Z. Wang and Q. Li, "Video quality assessment using a statistical model of human visual speed perception," Journal of the Optical Society of America A, vol. 24, no. 12, pp. B61-B69, Dec. 2007.